

## 1. Function and Use.

This small program will convert SJIS encoded Japanese characters into a ‘preprocessed’ form. The need of this program arises from the fact that this encoding uses the characters ‘\’, ‘{’, and ‘}’ which have special meanings in T<sub>E</sub>X.

Use this program as a filter:

```
sjisconv < input_file > output_file
```

## 2. The program.

The only function of this program is to replace all occurrences of SJIS encoded two byte characters **XY** with `^7fX^7fZZZ^7f` (**X** and **Y** are the first and the second byte of the character; **ZZZ** represents the second byte as a decimal number).

Additionally we define a `TEX` macro at the very beginning to signal a preprocessed file.

The following code is very simple. No error detection is done because `TEX` which will see the output of `sjisconv` complains loudly if something is wrong.

```
#define banner "sjisconv_(CJK_ver._4.7.0)"
#include <stdio.h>
#include <stdlib.h>

int main(argc, argv)
    int argc;
    char *argv[];
{int ch;
  fprintf(stdout, "\\def\\CJKpreproc{%s}", banner);
  ch = fgetc(stdin);
  while (!feof(stdin))
    {if ((ch ≥ #81 ∧ ch ≤ #9F) ∨ (ch ≥ #E0 ∧ ch ≤ #EF))
      {fprintf(stdout, "\\177%c\\177", ch);
        ch = fgetc(stdin);
        if (!feof(stdin))
          fprintf(stdout, "%d\\177", ch);
        }
      else
        fputc(ch, stdout);
      ch = fgetc(stdin);
    }
  exit(EXIT_SUCCESS);
  return 0;
}
```

/\* never reached \*/